CONTENT-DISTRIBUTION NETWORKS

George Porter
June 1, 2018

ATTRIBUTION

• These slides are released under an Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0) Creative Commons license
• These slides incorporate material from:
  • Kyle Jamieson, Princeton University (also under a CC BY-NC-SA 3.0 Creative Commons license)
ANNOUNCEMENTS

OUTLINE

1. Web caching
2. Content-distribution networks
   • Featuring Akamai
   • Overlay networks
WEB CACHING

- Many clients transfer the **same information**
- Generates **redundant** server and network load
- Also, clients may experience high **latency**

WHY WEB CACHING?

- Motivation for **placing content closer to client**:
  - User gets **better response time**
    - Content providers get happier users
  - Network gets **reduced load**
  - Why does caching work? Exploits locality of reference
  - How well does caching work?
    - Very well, **up to a limit**
    - Large overlap in content
    - But many unique requests
CACHING WITH REVERSE PROXIES

- Cache data close to origin server → decrease server load
- Typically done by content providers
- Client thinks it is talking to the origin server (the server with content)
- Does not work for dynamic content

CACHING WITH FORWARD PROXIES

- Cache close to clients → less network traffic, less latency
- Typically done by ISPs or corporate LANs
- Client configured to send HTTP requests to forward proxy
- Reduces traffic on ISP-1’s access link, origin server, and backbone ISP
CACHING & LOAD-BALANCING: OUTSTANDING PROBLEMS

• Problem ca. 2002: How to reliably deliver large amounts of content to users worldwide?

• Popular event: “Flash crowds” overwhelm (replicated) web server, access link, or back-end database infrastructure

• More rich content: audio, video, photos

• Web caching: Diversity causes low cache hit rates (25–40%)

OUTLINE

1. Web caching
2. Content-distribution networks
   • Featuring Akamai
   • Overlay networks
CONTENT DISTRIBUTION NETWORKS

• **Proactive content replication**
  • Content provider (*e.g.* CNN) pushes content out from its own *origin server*

• **CDN replicates** the content
  • On many servers spread throughout the Internet
  • Updating the replicas
    • Updates **pushed to replicas** when the content changes

REPLICA SELECTION: GOALS

• **Live** server
  • For availability

• **Lowest** load
  • To balance load across the servers

• **Closest**
  • Nearest geographically, or in round-trip time

• **Best** **performance**
  • Throughput, latency, reliability...
AKAMAI STATISTICS

- Distributed servers
  - Servers: ~100,000
  - Networks: ~1,000
  - Countries: ~70
- Many customers
  - Apple, BBC, FOX, GM
  - IBM, MTV, NASA, NBC, NFL, NPR, Puma, Red Bull, Rutgers, SAP, ...
- Client requests
  - 20+M per second
  - Half in the top 45 networks
  - 20% of all Web traffic worldwide

HOW AKAMAI USES DNS

- GET
- "index.html"
- cache.cnn.com/"foo.jpg"
- Akamai global DNS server
- Akamai regional DNS server
- Nearby Akamai cluster
- End user
- HTTP
- cnn.com (content provider)
- DNS TLD server
- Akamai cluster
HOW AKAMAI USES DNS

1. DNS lookup for cache.cnn.com
2. Akamai global DNS server
3. Akamai regional DNS server
4. Nearby Akamai cluster
5. End user
6. ALIAS: a73.g.akamai.net

HTTP

cnn.com (content provider)

DNS TLD server

HOW AKAMAI USES DNS

cnn.com (content provider)    DNS TLD server

1  2  3  4  5  6  7  8

Akamai global DNS server
Akamai regional DNS server
Nearby Akamai cluster

Address 1.2.3.4

End user

GET /foo.jpg
Host: cache.cnn.com
HOW AKAMAI WORKS: CACHE HIT

MAPPING SYSTEM

• Equivalence classes of IP addresses
  • IP addresses experiencing similar performance
  • Quantify how well they connect to each other

• Collect and combine measurements
  • Ping, traceroute, BGP routes, server logs
    • e.g., over 100 TB of logs per days
  • Network latency, loss, throughput, and connectivity
ROUTING CLIENT REQUESTS WITH THE MAP

- Map each **IP class** to a preferred **server cluster**
  - Based on performance, cluster health, etc.
  - Updated roughly every minute
    - **Short, 60-sec DNS TTLs** in Akamai regional DNS accomplish this
- Map client request to a server in the cluster
  - **Load balancer** selects a specific server
  - *e.g.*, to **maximize** the **cache hit rate**

ADAPTING TO FAILURES

- Failing **hard drive** on a server
  - Suspends after finishing “in progress” requests
- Failed **server**
  - Another server takes over for the IP address
  - Low-level map updated **quickly** (load balancer)
- Failed **cluster**, or **network path**
  - High-level map updated **quickly** (ping/traceroute)
TAKE-AWAY POINTS: CDNS

- Content distribution is hard
  - Many, diverse, changing objects
  - Clients distributed all over the world

- Moving content to the client is key
  - Reduces latency, improves throughput, reliability

- Content distribution solutions evolved:
  - Load balancing, reactive caching, to
  - Proactive content distribution networks

OUTLINE

1. Web caching
2. Content-distribution networks
   - Featuring Akamai
   - Overlay networks
ABSTRACTIONS AND OVERLAYS

OVERLAYS FOR ROUTING: WHY?

- Triangle inequality doesn’t hold in networks!
FORWARDING TRAFFIC THROUGH TUNNELING

OVERLAY NETWORKS FOR ROUTING

- Underlying network
- Internet connectivity (IP Routing)
OVERLAY NETWORKS

• Potential overlay connectivity
• SF as root

OVERLAY NETWORKS

• Determine edge weights
• E.g., bandwidth, latency
OVERLAY NETWORKS

- Build overlay connectivity
- An application-layer distribution tree